Rapport de projet

Site de généalogie participative

Réalisé par

Marlène GUILLEMETTE et Morgane VIDAL

Sous la direction de

Alain SABATIER

Pour l'obtention du DUT informatique

Année universitaire 2012-2013

Remerciements

Nous tenons à remercier M. Sabatier de nous avoir aidées et conseillées tout au long de la réalisation de notre projet, Mme Mami pour l'aide qu'elle nous a fournie pour la réalisation des diagrammes et Mme Gelsomino pour nous avoir aidées à réaliser le rapport de projet.

Nous remercions également M. Gaëtan Chabert pour l'aide qu'il nous a fournie pendant le développement de notre site web.

Enfin, nous remercions nos proches pour leur participation à la phase de test du site.

Sommaire

1. Cahier des charges ..... - 2 -
1.1. Sujet ..... - 2 -
1.2. Besoins généraux des utilisateurs ..... - 2 -
1.3. Analyse des besoins fonctionnels ..... - 2 -
1.4. Analyse des besoins non fonctionnels ..... - 3 -
2. Rapport technique ..... - 5 -
2.1. L'architecture du site ..... - 5 -
2.2. L'enregistrement des données ..... - 5 -
2.3. Les arbres ..... - 8 -
2.4. La recherche ..... - 11 -
2.5. Mon profil ..... - 12 -
2.6. La rubrique d'aide ..... - 13 -
3. Manuel d'utilisation ..... - 14 -
3.1. L'inscription ..... - 14 -
3.2. Connexion ..... - 15 -
3.3. Mon arbre ..... - 16 -
3.4. La recherche ..... - 18 -
3.5. Gestion du profil ..... - 19 -
3.6. Aide ..... - 20 -
4. Rapport d'activité ..... - 21 -
4.1. Cycle de développement ..... - 21 -
4.2. Planification ..... - 22 -
4.3. Méthodes et outils de travail ..... - 22 -

Table des figures

Figure 1 - Arborescence des fichiers ..... - 5 -

Figure 2 - Diagramme de classes..... - 6 -

Figure 3 - Table Individu..... - 7 -

Figure 4 - Table Users ..... - 7 -

Figure 5 - Diagramme UseCase d'un utilisateur voulant modifier son arbre ..... - 8 -

Figure 6 - Diagramme d'activités de la modification des données d'un individu dans la base de données..... - 9 -

Figure 7 - Tracé de traits ..... - 10 -

Figure 8 - Fonction is_valide_date() ..... - 10 -

Figure 9 - Impression d'écran d'une erreur lors de la modification des données d'un individu..... - 10 -

Figure 10 - Fonction RechercheArbre() ..... - 11 -

Figure 11 - Traitement du formulaire de contact..... - 13 -

Figure 12 - Inscription, 1re étape ..... - 14 -

Figure 13 - Inscription, 2e étape..... - 14 -

Figure 14 - Inscription réussie..... - 14 -

Figure 15 - Connexion, 1re étape ..... - 15 -

Figure 16 - Connexion, 2e étape..... - 15 -

Figure 17 - Connexion réussie..... - 15 -

Figure 18 - Formulaire pour ajouter une personne à l'arbre ..... - 16 -

Figure 19 - Modification d'une personne..... - 16 -

Figure 20 - Suppression d'un individu..... - 17 -

Figure 21 - Impression de l'arbre..... - 17 -

Figure 22 - Formulaire de recherche ..... - 18 -

Figure 23 - Résultats de la recherche ..... - 18 -

Figure 24 - Mon profil..... - 19 -

Figure 25 - Tableau récapitulatif des informations personnelles ..... - 19 -

Figure 26 - Gestion profil..... - 19 -

Figure 27 - Organisation du développement ..... - 21 -

Glossaire

Les termes définis dans ce glossaire sont identifiables dans le corps du texte au moyen d'un astérisque (*).

Généalogie participative : Les utilisateurs du site pourront créer leur arbre et consulter les arbres des autres utilisateurs s'ils en ont la permission.

Design pattern : Recueil de bonnes pratiques de conception pour un certain nombre de problèmes récurrents en programmation orientée objet.

Introduction

De nos jours, le nombre de Français s'intéressant à leur généalogie est en constante augmentation. La généalogie est la pratique qui a pour objet la recherche de l'origine et des liens entre les personnes et les familles. Sur internet, nombreux sont les sites et logiciels libres permettant de créer son propre arbre généalogique. Ces outils proposent à l'utilisateur d'enregistrer les membres de sa famille dans son arbre.

Ces sites et logiciels sont parfois difficiles d'utilisation et ne permettent généralement pas de partager l'arbre créé et de consulter les arbres des autres membres. Remédier à ces problèmes représente donc le cœur de notre projet.

Le projet que nous avons réalisé cherche, à travers la création d'un nouveau site de généalogie participative et le développement de fonctionnalités plus poussées propres à notre site, à améliorer la structure et l'utilisation des outils déjà disponibles sur le net.

Dans une première partie, nous décrirons la conception du cahier des charges qui regroupe les besoins des utilisateurs et les objectifs du site. Puis nous présenterons le rapport technique dans lequel toutes les étapes de la conception et du développement sont explicitées. De ce fait, nous montrerons les différents choix que nous avons effectués, les difficultés techniques que nous avons rencontrées, et les erreurs conceptuelles que nous avons du corriger par la suite. Ensuite, nous présenterons le manuel d'utilisation adressé aux futurs utilisateurs de notre site web. Et enfin, nous rendrons compte de la planification et de l'organisation à travers le rapport d'activité.

1. Cahier des charges

1.1. Sujet

« Mettre en place un site de généalogie capable de stocker plusieurs familles, d'imprimer des arbres sous plusieurs formats et de faire des rapprochements éventuels entre des familles différentes sur des ancêtres communs. »

1.2. Besoins généraux des utilisateurs

Les utilisateurs du site pourront soit être connectés (on les appelle alors des membres), soit naviguer sur le site sans connexion préalable, on les appelle alors des invités. Il n'y a pas d'administrateur donc pas de back-office sur le site.

1.2.1. Besoins de l'invité

Les invités pourront s'inscrire sur le site et donc se connecter/déconnecter du site.

1.2.2. Besoins des membres

Les membres auront la possibilité de rechercher facilement une autre personne inscrite sur le site, si celle-ci a accepté qu'on regarde son arbre. Lorsqu'ils auront trouvé la personne qu'ils recherchaient, ils pourront consulter leur arbre.

Chaque membre devra pouvoir accéder à son arbre, pouvoir le modifier et pouvoir l'imprimer.

1.3. Analyse des besoins fonctionnels

1.3.1. Fonctionnalités communes

Chaque personne s'inscrivant sur le site renseignera son prénom, son nom (nom de naissance et nom actuel), sa profession (optionnel), sa date de naissance, son sexe, son adresse mail et enfin son mot de passe. Il devra également choisir quel paramètre de confidentialité il préfère (soit son arbre généalogique sera visible par les autres utilisateurs, soit il sera privé).

Chaque utilisateur pourra préciser qui est son père et sa mère, et donc par extension, préciser les parents des membres de sa famille. L'arbre se construira donc au fur et à mesure des ajouts que fait l'utilisateur. Certaines informations sont nécessaires pour ajouter une personne à son arbre : le prénom de la personne, son nom de naissance et son sexe. D'autres informations sont optionnelles : le nom de famille actuel de la personne, sa profession, sa date de naissance, de décès.

1.3.2. Besoins de l'invité

Pour un invité, il y aura deux menus présents sur toutes les pages. Le premier menu s'agrandira lorsque vous serez connecté. En tant qu'invité vous ne verrez qu'une rubrique :

Accueil : Cette rubrique présentera brièvement le site. Elle comprend un lien vers le formulaire d'inscription et vers celui de connexion.

Le deuxième menu propose deux rubriques si l'internaute est invité :

Plan du site : Cette rubrique permettra à l'internaute d'avoir un aperçu de la totalité des pages du site. Si la personne visitant le site n'est pas connectée, elle aura accès aux pages pour s'inscrire et se connecter.

Aide : Cette rubrique mettra à disposition des utilisateurs des moyens de contacter l'administrateur du site. Elle permettra éventuellement aux utilisateurs de télécharger le manuel d'utilisation.

1.3.3. Besoins des membres

Lorsque l'internaute se sera connecté, plusieurs rubriques se rajouteront au premier menu :

Recherche : Cette rubrique permettra aux utilisateurs de rechercher d'autres personnes sur le site et de consulter leur arbre si celles-ci ont accepté de le partager.

Mon arbre : Cette rubrique permet de consulter, éditer, et imprimer son arbre généalogique.

Dans le deuxième menu, une autre rubrique apparaîtra :

Mon profil : Cette rubrique vous permettra de modifier vos informations personnelles, vous déconnecter et enfin supprimer votre compte.

Le site fera des rapprochements éventuels entre les familles inscrites sur le site grâce à la page recherche qui, pour un nom donné, renvoie toutes les personnes inscrites sur le site ayant un nom similaire. Le membre pourra ensuite consulter son arbre.

1.4. Analyse des besoins non fonctionnels

1.4.1. Spécifications techniques

Nous avons décidé d'utiliser WinDesign et Modelio afin de réaliser les diagrammes. Pour le calendrier prévisionnel, nous avons utilisé GanttProject qui est un logiciel qui permet de faire des diagrammes de Gantt.

Pour communiquer entre nous et discuter du projet nous avons décidé d'utiliser Skype. Nous avons également utilisé nos messageries institutionnelles afin d'échanger des pages de code.

Nous ferons le code de notre site en utilisant plusieurs langages : HTML, PHP, mySQL et CSS. Nous utiliserons éventuellement quelques fonctions en JavaScript. Notre site sera entièrement réalisé en suivant le design pattern MVC.

Notre site sera conforme à la norme W3C. W3C est un site qui valide la construction des pages web.

1.4.2. Aspects ergonomiques

Pour plus de fluidité dans la navigation, l'utilisateur pourra retourner à la page d'accueil à partir de n'importe quelle page.

Notre site web sera bien sûr accessible à tous les internautes et sera donc par conséquent facile d'utilisation pour permettre aux personnes de tout âge de s'en servir.

2. Rapport technique

2.1. L'architecture du site

2.1.1. Conception

Nous avons décidé de construire notre site en suivant une architecture MVC. MVC (modèle vue contrôleur) est un design pattern* qui permet de séparer les données, leur présentation et leur traitement. C'est donc un modèle qui permet de mieux organiser le code du site.

Nous avons choisi de trier les pages de code dans trois dossiers différents : Modèle, Vue et Contrôleur.

2.1.2. Résultat

Controleur 29/12/2012 18:41 File folder
Download 29/12/2012 18:41 File folder
Image 29/12/2012 18:41 File folder
Modele 29/12/2012 18:41 File folder
Vue 29/12/2012 18:41 File folder
connection.php 29/12/2012 18:41 PHP File 1 KB
footer.php 29/12/2012 18:41 PHP File 1 KB
print.css 29/12/2012 18:41 CSS File 2 KB
style.css 29/12/2012 18:41 CSS File 5 KB

Figure 1 - Arborescence des fichiers

Dans les dossiers Controleur, Modele, et Vue nous avons respectivement les contrôleurs, les modèles et les vues. Dans le dossier Download il y a le manuel de l'utilisateur qui est téléchargeable sur le site dans la rubrique « Aide ». Le dossier Image comprend toutes les images qui apparaissent sur le site.

Le fichier « connexion » n'est pas dans un des dossiers cités avant, car il contient le script permettant de se connecter à la base de données. Le fichier « footer » est le pied de page de toutes les pages du site. Les deux pages de style print et style sont des fichiers au format css. La feuille de style s'intitulant « style » permet la mise en forme du texte et des images sur tout le site. La feuille de style « print » va servir uniquement lors de l'impression de l'arbre généalogique.

2.2. L'enregistrement des données

2.2.1. Conception

Le diagramme de classes de notre projet est relativement simple, puisque nous avons à la fin seulement deux classes. Comme le montre la figure 2 ci-dessous, la classe Individu a

tous les attributs nécessaires pour enregistrer une personne dans un arbre généalogique. La classe Utilisateur, elle, va servir à enregistrer les données liées aux personnes s'inscrivant sur le site. La classe Utilisateur hérite de la classe Individu puisqu'un utilisateur va aussi apparaître dans un arbre généalogique et donc avoir les mêmes attributs que les autres membres de l'arbre.

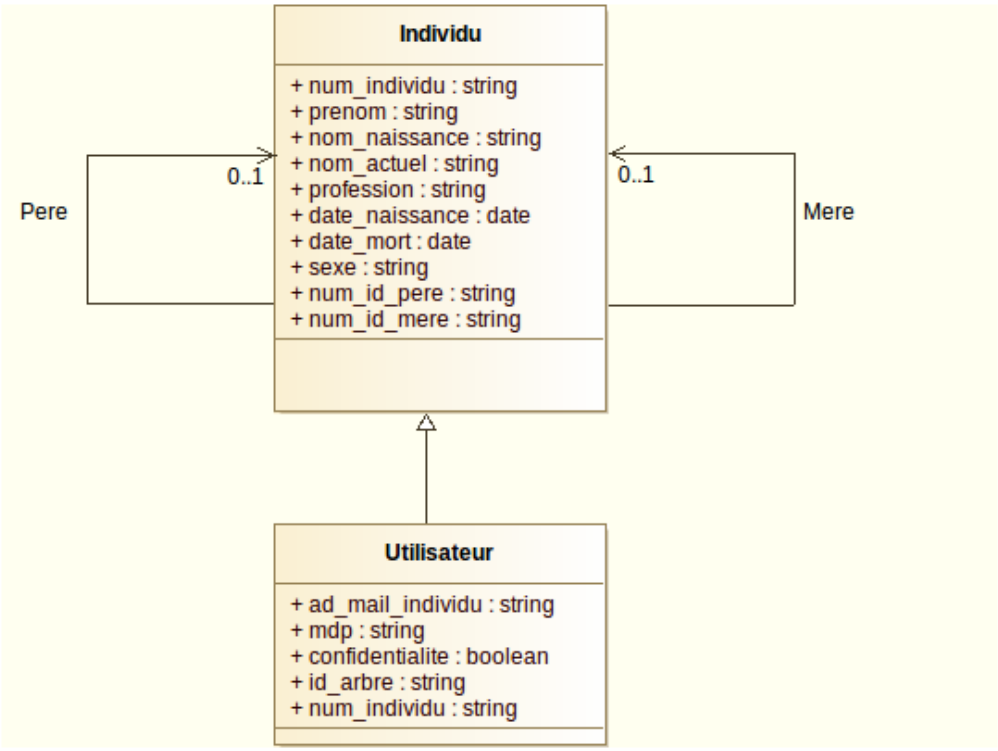


Figure 2 - Diagramme de classes

Un individu est caractérisé par un numéro d'individu qui est donc la clé primaire de cette table. Le num_individu est généré par la fonction uniqid(), et permet d'être sûr que tous les individus ont un identifiant unique. Le num_id_pere et le num_id_mere sont les identifiants du père et de la mère de l'individu.

Un utilisateur est défini par son adresse mail. En effet, il est impossible de créer plusieurs comptes avec la même adresse. Pour des raisons de sécurité, nous ne stockons pas les mots de passe en clair dans notre base de données. Nous avons choisi de les crypter avant de les enregistrer dans la base de données grâce à la fonction md5(). L'attribut confidentialité est un booléen puisque l'utilisateur n'a le choix qu'entre deux options : rendre son arbre visible aux autres utilisateurs, ou le garder privé. Enfin, l'id_arbre correspond à l'identifiant de l'arbre de l'utilisateur. Ainsi dans la table utilisateur on a accès à tous les identifiants des arbres. Le num_individu dans la table Utilisateur est une clé étrangère. Elle permet de faire le lien entre l'utilisateur et l'individu.

2.2.2. Résultat

Pour manipuler les données, nous avons donc créé deux tables dans notre base de données. Ces deux tables sont visibles dans les figures suivantes (3 et 4).

	Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
	num_individu	varchar(200)	latin1_swedish_ci		Non	Aucun		
	prenom	varchar(70)	latin1_swedish_ci		Non	Aucun		
	nom_naissance	varchar(70)	latin1_swedish_ci		Non	Aucun		
	profession	varchar(150)	latin1_swedish_ci		Non	Aucun		
	nom_actuel	varchar(70)	latin1_swedish_ci		Non	Aucun		
	date_naissance	date			Non	Aucun		
	date_mort	date			Non	Aucun		
	sexe	varchar(10)	latin1_swedish_ci		Non	Aucun		
	num_individu_pere	varchar(200)	latin1_swedish_ci		Non	Aucun		
	num_individu_mere	varchar(200)	latin1_swedish_ci		Non	Aucun		

Figure 3 - Table Individu

	Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
	admailindividu	varchar(70)	latin1_swedish_ci		Non	Aucun		
	mdp	varchar(50)	latin1_swedish_ci		Non	Aucun		
	confidentialite	tinyint(1)			Non	Aucun		
	id_arbre	varchar(200)	latin1_swedish_ci		Non	Aucun		
	num_individu	varchar(200)	latin1_swedish_ci		Non	Aucun		

Figure 4 - Table Users

2.3. Les arbres

2.3.1. Conception

Un utilisateur peut consulter son arbre, modifier des membres précédemment ajoutés sur l'arbre, les supprimer, en ajouter et enfin imprimer son arbre.

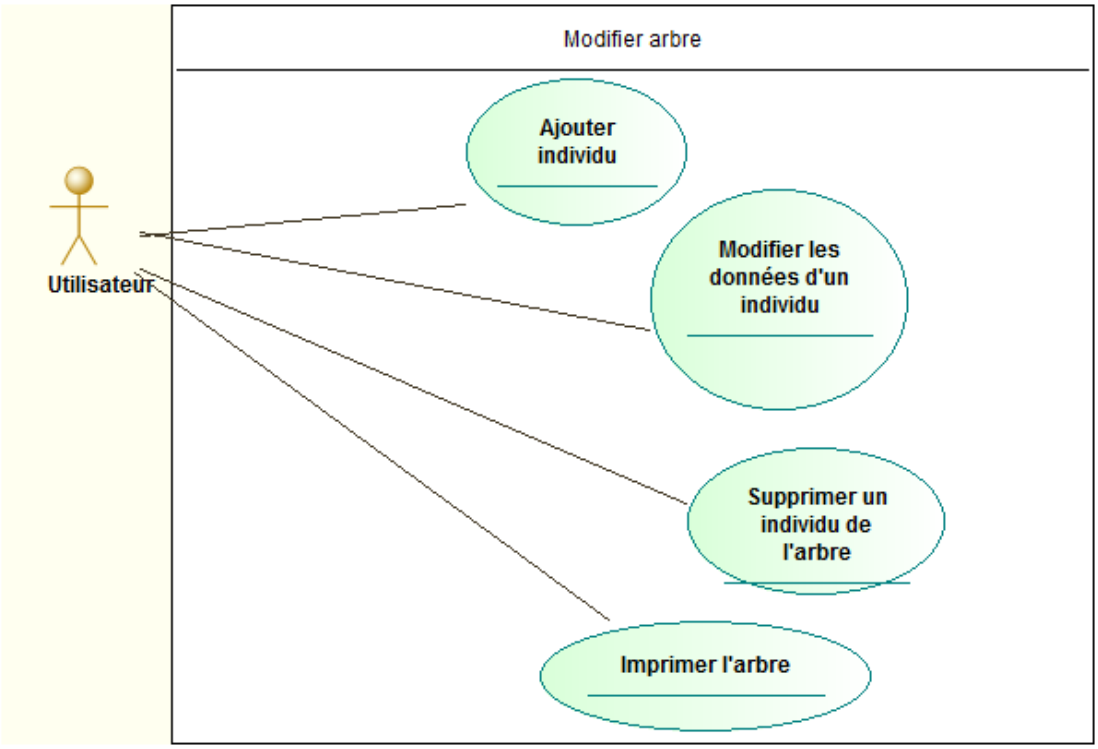


Figure 5 - Diagramme UseCase d'un utilisateur voulant modifier son arbre

Pour un individu, il y a deux ascendants directs : son père et sa mère. Chaque individu a donc dans ses attributs le numéro de sa mère et le numéro de son père. Lorsqu'on veut afficher l'arbre d'une personne, il suffit donc de prendre la racine de cet arbre, c'est-à-dire la personne dont on veut voir l'arbre. Si son père (ou sa mère) a été créé, on affiche ses informations. Sinon, on affiche un lien vers un formulaire pour ajouter cette personne à l'arbre.

Pour insérer un individu à l'arbre, il faut remplir un formulaire. Avant d'enregistrer les données saisies par l'utilisateur, on procède à plusieurs vérifications des données. On va d'abord vérifier que les champs obligatoires ont été remplis, soit le prénom et le nom. Ensuite on vérifie la cohérence des données : les dates doivent être au format valide (AAAA-MM-JJ). Ensuite, on enregistre l'individu dans la base de données.

Lors de la modification d'un des individus de l'arbre, on veut vérifier que l'utilisateur n'a pas inséré des données aberrantes ou oublié de remplir des champs obligatoires. Comme

vous pouvez le constater dans la Figure 6, si ces deux conditions ne sont pas validées, on renvoie l'utilisateur sur le formulaire pour qu'il procède à une correction.

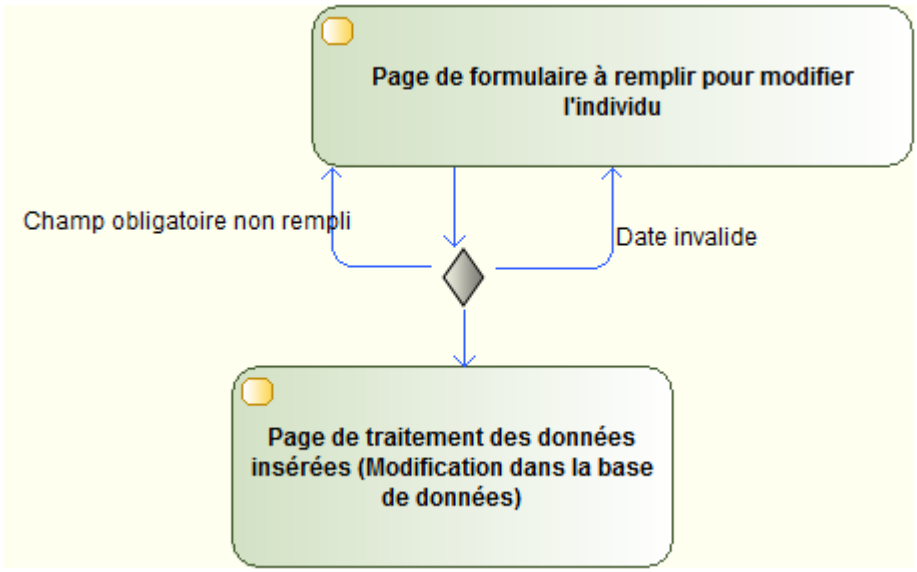

Page de formulaire à remplir pour modifier l'individu
Champ obligatoire non rempli
Date invalide
Page de traitement des données insérées (Modification dans la base de données)

Figure 6 - Diagramme d'activités de la modification des données d'un individu dans la base de données

Pour permettre aux utilisateurs de supprimer un des membres de leur famille de leur arbre généalogique, nous avons du créer un algorithme récursif. En effet, lorsqu'on supprime une personne, il est logique de supprimer tous ses ancêtres en même temps. La logique est la suivante : si la personne à supprimer possède un père ou une mère enregistrés dans la base de données, alors on va appliquer la fonction « supprimer » à ses ancêtres. Lorsqu'il n'y a plus d'ancêtres à supprimer, alors on supprime la racine de l'arbre, c'est-à-dire la personne qu'on voulait supprimer à la base. Sinon, si la personne n'a pas d'ancêtres, on supprime uniquement cette personne de la base de données.

Pour créer le bouton permettant l'impression de l'arbre, nous avons du créer une nouvelle feuille de style qui s'applique lors de l'impression. Cette feuille de style permet de définir quels éléments ne doivent pas être imprimés en les rendant invisibles. Par exemple, les liens pour modifier ou supprimer une des personnes de l'arbre ne devront pas être imprimés.

2.3.2. Résultat

Dans l'annexe 1, on peut voir le contrôleur de la page mon arbre. Le contrôleur sert à instancier les variables. Par exemple, si la racine de l'arbre possède un père, alors on va récupérer toutes les informations dans la base de données sur ce père. Ensuite, dans la vue (annexe 2), nous afficherons ses données. Si le père n'a pas été instancié dans le contrôleur, alors s'affiche un lien vers un formulaire pour ajouter son père à son arbre. Pour les traits entre les membres de l'arbre, nous avons utilisé l'élément canvas proposé par html5. Cet élément peut contenir des éléments graphiques qu'il faut ensuite tracer à l'aide des méthodes en JavaScript.

```
/*les traits vers "mon pere"*/
context.beginPath();
context.moveTo(110, 578);
context.lineTo(260, 578);
context.stroke();

context.beginPath();
context.moveTo(110, 578);
context.lineTo(110, 975);
context.stroke();
```

Figure 7 - Tracé de traits

Pour tracer un trait par exemple, on commence par dire que l'élément qu'on va tracer va être un trait grâce à l'instruction context.beginPath(). Ensuite on précise le point d'origine du trait avec l'instruction context.moveTo(x, y) et le point d'arrivée du trait avec context.lineTo(x, y). Les variables x et y sont les coordonnées des points par rapport à la page. L'instruction context.stroke() va tracer le trait précédemment défini.

Dans notre cas, nous avons du tracer un trait vers chacun des membres devant être affichés dans l'arbre.

Pour vérifier la validité d'une date, nous avons du créer une fonction spécifique. Cette fonction est visible dans la figure ci-dessous (figure 8). La fonction is_valide_date() prend en paramètres la date à valider. Pour vérifier si la date est valide, on va couper la date envoyée en trois éléments séparés par le séparateur « - ». Ces trois éléments sont l'année, le mois et le jour. On vérifie donc qu'il y a bien ces trois éléments dans la première boucle if. Ensuite, on va vérifier la validité de chacun de ces éléments en utilisant la fonction checkdate(). Cette fonction retournera faux si le mois est bien compris entre 1 et 12, si le jour est compris dans l'ensemble des jours du mois correspondant, et si l'année est bien valide.

```
207 //fonction qui permet de vérifier que la date insérée est valide
208 public static function is_valide_date($date, $sep='-')
209 {
210     if(!list($year, $month, $day) = explode($sep, $date))   //si en fonction du separateur et de la chaine $date on arrive pas à avoir une liste de la forme $year, $month, $day
211         {
212             return false;
213         }
214
215     return checkdate($month, $day, $year); //crée une date valide en fonction de $month, $year, $date
216 }
```

Figure 8 - Fonction is_valide_date()

Le site dispose de messages d'erreur spécifiques et ciblés. Par exemple, lorsqu'une personne veut modifier les données qu'elle avait enregistrées pour un de ces prédécesseurs, un message d'erreur ciblé la prévient si elle a oublié de remplir un champ. (Cf. figure 9)

Figure 9 - Impression d'écran d'une erreur lors de la modification des données d'un individu

Pour supprimer un individu d'un arbre, nous avons également créé une fonction visible dans l'annexe 3. Cette fonction va être récursive car on souhaite que la suppression d'un individu entraine la suppression de tous ses ancêtres : Tout d'abord, on va commencer par créer un objet individu à partir du num_individu de l'individu à supprimer afin de pouvoir lui appliquer des fonctions propres à cette classe. Ensuite, on va regarder si la personne qu'on veut supprimer possède des parents qui ont été créés et enregistrés par l'utilisateur. S'il en possède, on veut également les supprimer. C'est pourquoi on va également leur appliquer la fonction supprimer(). Après avoir fait cela, dans la première requête, on supprime l'individu de la base de données. Ensuite on va réinitialiser les attributs num_individu_pere et num_individu_mere de l'enfant de l'individu supprimé.

L'impression de l'arbre se fait grâce à la fonction window.print() qui est une méthode JavaScript. Nous avons créé une feuille de style, « print.css » qui est rendue spécifique pour le rendu de l'imprimante grâce à l'attribut media='print'. Dans cette feuille de style nous avons décidé d'afficher uniquement l'arbre (sans les liens pour ajouter, supprimer ou modifier les membres). Nous avons donc du cacher les liens, le header et le bouton d'impression grâce au paramètre display mis à la valeur none.

2.4. La recherche

2.4.1. Conception

La page de recherche a pour but de permettre aux utilisateurs de rechercher les ancêtres qu'ils pourraient avoir en commun avec les autres utilisateurs. Pour ce faire, il a été nécessaire de faire une fonction qui recherche des noms semblables à celui saisi par le membre dans la base de données en tenant compte des paramètres de confidentialité des utilisateurs.

2.4.2. Résultat

Après avoir récupéré le nom saisi par l'utilisateur à l'aide d'un formulaire, on utilise une méthode RechercheArbre() qui, pour un nom donné, va nous renvoyer un tableau des individus dont le nom correspond au nom recherché (ou en est composé).

```
148     //Méthode qui, pour un nom donné, va récuperer dans la bd tous les utilisateurs ayant un nom semblable.
149 public static function RechercheArbre($nomcherche){
150     $req =  "SELECT i.num_individu as num_individu, i.nom_naissance as nom_naissance, i.nom_actuel as nom_actuel, i.prenom as prenom,
151                 i.profession as profession, i.date_naissance as date_naissance, i.sexe as sexe
152             FROM Individu as i
153             INNER JOIN Users as u
154             ON i.num_individu = u.num_individu
155             WHERE i.nom_naissance LIKE '%$nomcherche%' OR i.nom_actuel LIKE '%$nomcherche%' AND u.confidentialite = 0";
156
157     $res = mysql_query($req) or die('erreur requete');
158
159     $users = Array(); //création d'un tableau
160
161     //on utilise un tableau à plusieurs dimensions pour pouvoir récuperer les infos de tous les utilisateurs répondant à la recherche
162     while($tuple = mysql_fetch_array($res)){
163         $users[$tuple['num_individu']]['num_individu'] = $tuple['num_individu'];
164         $users[$tuple['num_individu']]['nom_naissance'] = $tuple['nom_naissance'];
165         $users[$tuple['num_individu']]['nom_actuel'] = $tuple['nom_actuel'];
166         $users[$tuple['num_individu']]['prenom'] = $tuple['prenom'];
167         $users[$tuple['num_individu']]['profession'] = $tuple['profession'];
168         $users[$tuple['num_individu']]['date_naissance'] = $tuple['date_naissance'];
169         $users[$tuple['num_individu']]['sexe'] = $tuple['sexe'];
170     }
171     return $users;
172 }
```

Figure 10 - Fonction RechercheArbre()

Cette méthode est visible dans figure ci-dessus, et, comme vous pouvez le voir, nécessite une requête SQL assez complexe. Cette fonction prend en paramètres le nom à rechercher dans la base de données. On va donc faire une jointure de la table Individu et Utilisateur afin de pouvoir récupérer comme résultat un tableau d'individus puisque nous voulons afficher à la fin les informations des individus obtenus par la recherche, et bien évidemment un lien vers leur arbre. La jointure se fera sur l'attribut num_individu. Dans le WHERE, nous avons utilisé LIKE pour obtenir dans les résultats toutes les personnes dont le nom est composé du nom recherché, et non pas uniquement ceux qui ont exactement le même nom. Il ne faut pas oublier la dernière condition que notre requête doit prendre en compte : le fait qu'on n'affichera que les personnes qui ont accepté que l'on partage leurs informations et leur arbre. C'est pourquoi dans la requête on voit qu'il est demandé que le paramètre de confidentialité de l'utilisateur soit réglé à « 0 ».

Ensuite on place les résultats obtenus par la requête dans un tableau à deux dimensions. La première dimension est le numéro de l'individu car on obtient plusieurs individus. La seconde dimension regroupe tous les attributs comme le nom de naissance, le nom marital, le prénom, la profession, la date de naissance et le sexe. La fonction va ensuite retourner le tableau.

Pour l'affichage des résultats, on n'aura plus qu'à lire toutes les valeurs retournées par cette fonction.

2.5. Mon profil

2.5.1. Conception

La rubrique « mon profil » est très utile pour les membres car elle leur permet de modifier leurs données personnelles, de se déconnecter et également de supprimer définitivement leur compte. La suppression de compte nécessite une suppression dans la table Utilisateurs, ainsi qu'une déconnexion. Il paraît logique que lorsqu'on supprime son compte, on se déconnecte également en même temps.

2.5.2. Résultat

Nous avons décidé de vous expliquer quelques morceaux de code propres aux fonctionnalités offertes par la rubrique « mon profil ».

Pour permettre au membre de se déconnecter, nous avons simplement utilisé session_destroy(). Cette méthode propre à PHP va supprimer les données associées à la session en cours.

Pour permettre la modification des données d'un membre, nous avons utilisé une simple requête SQL comportant un UPDATE qui permet de mettre à jour les données du membre en question.

Et enfin, pour la suppression définitive des comptes, nous avons utilisé une requête DELETE pour supprimer l'utilisateur de la table Utilisateur ainsi qu'un session_destroy() qui permet la déconnexion.

2.6. La rubrique d'aide

2.6.1. Conception

La rubrique d'aide est indispensable à notre site pour permettre à tous les utilisateurs, qu'ils soient invités ou membres, de télécharger le manuel d'utilisation et de pouvoir contacter le créateur du site (par mail) en cas de problème.

2.6.2. Résultat

Le fait de permettre le téléchargement du manuel d'utilisateur n'a pas été très difficile à gérer. En effet, il suffit de faire un lien vers le fichier placé dans le dossier « Download ». Le formulaire de contact est un formulaire basique. Lors du traitement du formulaire, on va envoyer le contenu saisi par l'internaute au créateur du site.

```php
<?php
    $titre = "Envoi mail";
    include("./header.php");

    if (!empty($_POST["nom"]) && !empty($_POST["email"]) && !empty($_POST["message"]))
    {
        $to="marlene.guillemette@etud.univ-montp2.fr";
        $sujet=$_POST["objet"];
        $corps=$_POST["message"];
        $headers = "From:" . $_POST["email"] . "\r\n" .
        "X-Mailer: PHP/" . phpversion();
        mail($to,$sujet,$corps,$headers);
        include ("../Vue/traitement_contact.php");
    }

    else{
        include("../Vue/formulaire_non_rempli.php");
    }

    include("../footer.php");
?>
```

Figure 11 - Traitement du formulaire de contact

On commence par vérifier que tous les champs ont bien été remplis. S'ils ont bien été remplis, on va préparer les éléments nécessaires pour utiliser la fonction mail. On définit le destinataire, le sujet du mail qui sera l'objet choisi par l'utilisateur lorsqu'il a rempli le formulaire de contact, le corps qui sera le message que l'utilisateur veut envoyer et enfin on définit tous les éléments additionnels à mettre dans l'en-tête du mail. On va préciser l'envoyer du mail dans chaque mail. L'instruction "X-Mailer: PHP/". phpversion(); ajoutée à l'entête permet d'éviter les filtres de spam.

Il suffit ensuite d'utiliser la méthode php mail() en précisant tous les paramètres nécessaires (ligne 23).

Si l'internaute n'a pas rempli les champs on lui envoie un message d'erreur.

3. Manuel d'utilisation

3.1. L'inscription

L'inscription sur notre site est accessible dés la page d'accueil. Pour vous inscrire, il suffit de cliquer sur « Inscrivez-vous ».

Plan du site
Aide
ACCUEIL
RECHERCHE
MON ARBRE
Bienvenue sur Généal
Généal est un site de généalogie participative créé par Vidal Morgane et Guillemette Marlène.
Ce site permet à ses utilisateurs de facilement faire leur arbre généalogique, et de consulter les arbres des autres utilisateurs. Si vous êtes perdu, vous pouvez accéder au plan du site ou à la page d'aide qui comprend un manuel d'utilisateur téléchargeable et un formulaire de contact pour répondre à toutes vos questions.
Vous êtes nouveau?
Inscrivez vous!
Déjà inscrit?
Connectez vous!

Figure 12 - Inscription, 1re étape

Ensuite, il faudra remplir le formulaire suivant. Attention, votre adresse mail, votre mot de passe, et le paramètre de confidentialité que vous choisirez ne seront pas modifiables.

Veillez aussi à bien remplir les champs obligatoires.

Figure 13 - Inscription, 2e étape

Si tout se passe bien, vous serez connecté au site sur votre nouveau compte.

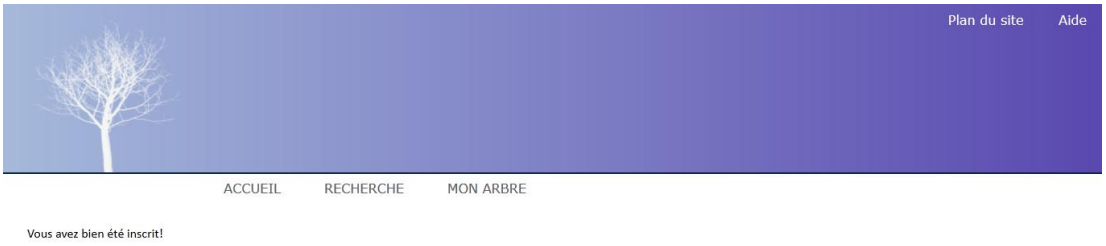


Figure 14 - Inscription réussie

3.2. Connexion

La connexion sur le site est également accessible dés la page d'accueil. Pour vous connecter, il suffit de cliquer sur « Connectez-vous ».

Plan du site
Aide
ACCUEIL
RECHERCHE
MON ARBRE
Bienvenue sur Généal
Généal est un site de généalogie participative créé par Vidal Morgane et Guillemette Marlène.
Ce site permet à ses utilisateurs de facilement faire leur arbre généalogique, et de consulter les arbres des autres utilisateurs. Si vous êtes perdu, vous pouvez accéder au plan du site ou à la page d'aide qui comprend un manuel d'utilisateur téléchargeable et un formulaire de contact pour répondre à toutes vos questions.
Vous êtes nouveau?
Inscrivez vous!
Déjà inscrit?
Connectez vous!

Figure 15 - Connexion, 1re étape

Il vous suffit ensuite de saisir votre adresse mail et votre mot de passe.

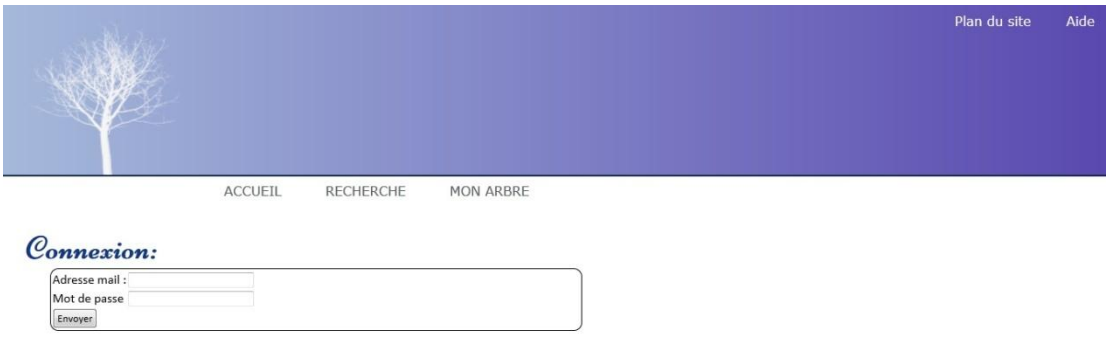


Figure 16 - Connexion, 2e étape

Si la connexion s'est bien passée, la page suivante (figure 17) s'affiche.

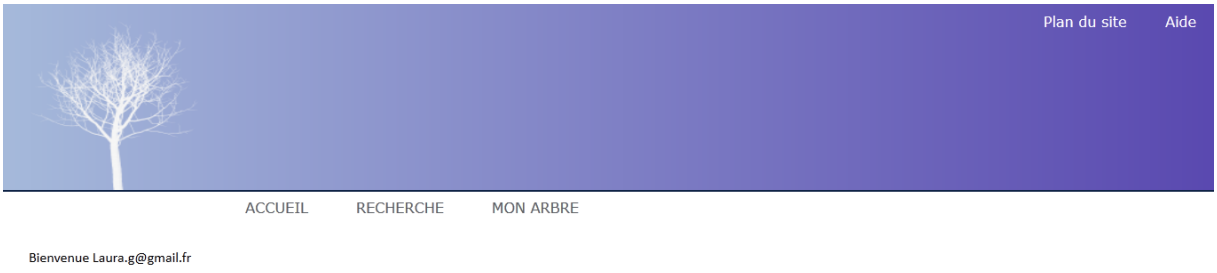


Figure 17 - Connexion réussie

Si cette fenêtre ne s'affiche pas, vous avez fait une erreur en saisissant votre adresse mail ou votre mot de passe. Un message d'erreur s'affichera et il vous faudra remplir à nouveau le formulaire de connexion.

3.3. Mon arbre

3.3.1. Ajouter une personne à mon arbre

Vous êtes inscrit et vous désirez maintenant pouvoir construire votre arbre généalogique ?

Vous aurez le choix entre commencer par ajouter votre père, ou votre mère. Pour cela, il faudra cliquer sur un des liens entourés en rouge dans l'annexe 4. Ces liens vous redirigeront vers un formulaire (cf. figure 18) qu'il vous faudra remplir. Les champs qui ne sont pas obligatoires pourront être remplis plus tard, lorsque vous aurez réussi à trouver les données qu'il vous manquait.

Figure 18 - Formulaire pour ajouter une personne à l'arbre

Après avoir cliqué sur « enregistrer », vous serez redirigé vers la page de votre arbre.

Il faudra procéder de cette façon pour ajouter tout les membres de votre famille à votre arbre.

3.3.2. Modifier un des membres de mon arbre

Pour modifier une des personnes de l'arbre, il faut cliquer sur « modifier cette personne » dans le cadre de la personne à modifier.

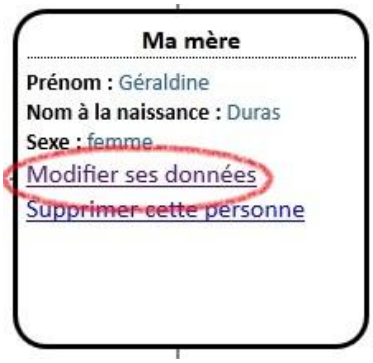


Figure 19 - Modification d'une personne

Vous serez dirigé sur un formulaire pré-rempli avec les données que vous aviez enregistré pour cet individu. Il ne vous restera plus qu'à corriger les valeurs ou rajouter celles qui manquent, puis vous pourrez enregistrer ces modifications en cliquant sur « enregistrer les modifications ». Après enregistrement, vous serez redirigé sur votre arbre.

3.3.3. Supprimer un des membres de l'arbre

Pour supprimer un individu de votre arbre, il vous faut consulter votre arbre en cliquant sur l'onglet « mon arbre ». Il vous suffira ensuite de cliquer sur le lien « supprimer cette personne » dans le cadre qui est propre à l'individu que vous souhaitez supprimer.

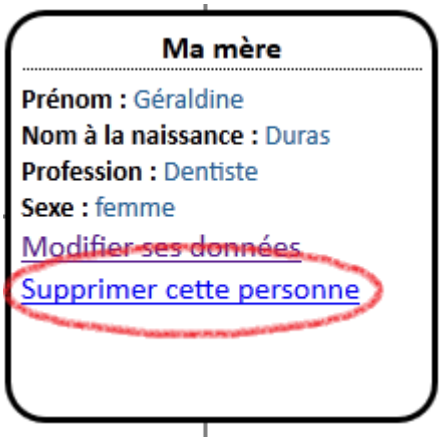


Figure 20 - Suppression d'un individu

3.3.4. Imprimer l'arbre

Pour imprimer votre arbre, il faut consulter votre arbre en cliquant sur l'onglet « mon arbre ». Ensuite, il vous suffit de cliquer sur le bouton « impression » situé en haut de la page. Vous pourrez ainsi régler les paramètres spécifiques à votre imprimante dans la petite fenêtre qui s'ouvrira.

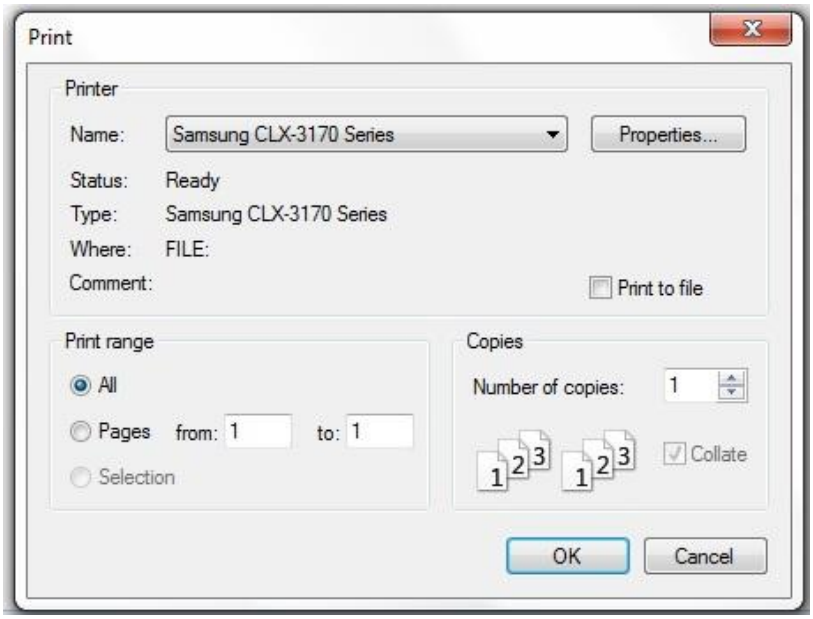


Figure 21 - Impression de l'arbre

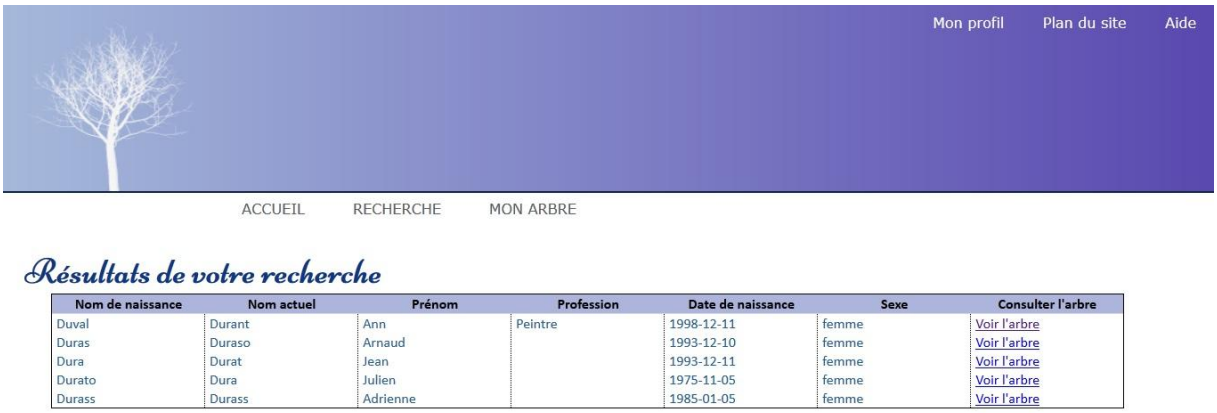
3.4. La recherche

Vous avez construit votre arbre et vous voulez voir si d'autres personnes n'ont pas un ancêtre commun avec vous ? Vous vous rappelez que votre grand-mère citait un certain « Adrienne Duras » mais vous voudriez avoir plus d'informations sur cette personne, même si vous n'êtes pas sûr de l'orthographe de son nom ?

Il vous faudra alors aller dans la rubrique « Recherche » et saisir le nom de famille que vous souhaitez rechercher.

Figure 22 - Formulaire de recherche

Après avoir cliqué sur « rechercher », un tableau contenant tous les résultats s'affichera.



Nom de naissance	Nom actuel	Prénom	Profession	Date de naissance	Sexe	Consulter l'arbre
Duval	Durant	Ann	Peintre	1998-12-11	femme	Voir l'arbre
Duras	Duraso	Arnaud		1993-12-10	femme	Voir l'arbre
Dura	Durat	Jean		1993-12-11	femme	Voir l'arbre
Durato	Dura	Julien		1975-11-05	femme	Voir l'arbre
Durass	Durass	Adrienne		1985-01-05	femme	Voir l'arbre

Figure 23 - Résultats de la recherche

Il vous suffira alors de cliquer sur « voir l'arbre » pour consulter l'arbre de chaque personne trouvée. Dans l'annexe 5, vous trouverez une impression d'écran d'un exemple d'arbre consulté suite à une recherche.

3.5. Gestion du profil

3.5.1. Consulter mon profil

Pour consulter votre profil, il vous faudra cliquer sur l'onglet « mon profil » situé en haut à droite de toutes les pages du site.

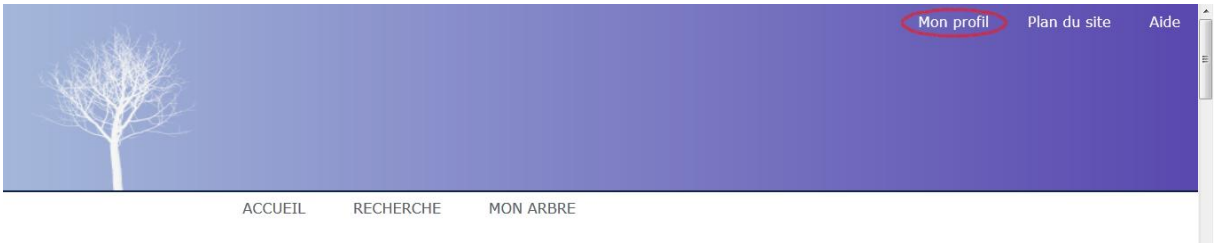


Figure 24 - Mon profil

Ensuite vous pourrez consulter un tableau récapitulatif de toutes vos données personnelles.

Figure 25 - Tableau récapitulatif des informations personnelles

3.5.2. Modifier ses données personnelles

Pour modifier vos données personnelles, il vous faudra utiliser le lien situé juste en dessous du tableau récapitulatif.

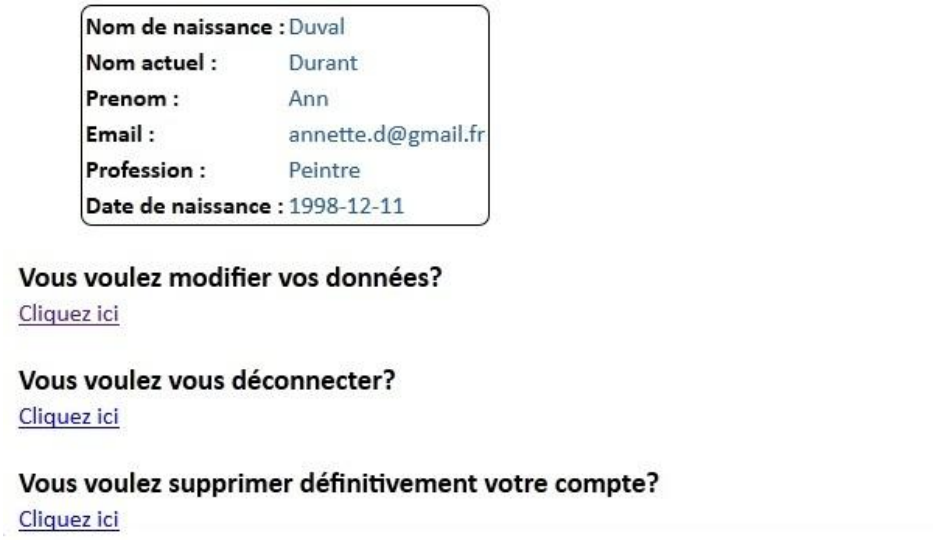


Figure 26 - Gestion profil

Dans la figure 26, vous pouvez voir que c'est le premier lien en dessous du tableau. Lorsque vous aurez cliqué sur ce lien, vous accèderez à un formulaire pré rempli avec vos données actuelles. Vous n'aurez plus qu'à modifier les données incorrectes puis à cliquer sur « enregistrer les modifications ».

3.5.3. Se déconnecter

Pour se déconnecter, il suffit de cliquer sur le deuxième lien en dessous du tableau (cf. figure 26). Après avoir cliqué sur ce lien, vous serez déconnecté et vous aurez un statut « d'invité » sur le site. Il faudra vous reconnecter pour pouvoir accéder à votre arbre, votre profil et la page de recherche.

3.5.4. Supprimer mon compte

Pour supprimer votre compte, il vous suffira de cliquer sur le troisième et dernier lien de la page de votre profil (cf. figure 26). Lorsque vous cliquez sur le lien, la suppression de votre compte se fait automatiquement. Attention, cette opération est irréversible.

3.6. Aide

3.6.1. Le manuel de l'utilisateur

Le manuel de l'utilisateur est téléchargeable sur le site dans la rubrique « Aide ». Lorsque vous cliquerez sur « Télécharger » le manuel s'ouvrira sous forme de PDF.

3.6.2. Le plan du site

Le sitemap est accessible dans la rubrique « plan du site ». Ce plan du site comporte des liens vers toutes les pages du site.

3.6.3. Le formulaire de contact

Si malgré ce manuel et le plan du site vous n'avez pas trouvé de solution à vos problèmes vous pouvez contacter le créateur du site en remplissant le formulaire présent dans la rubrique d'aide.

4. Rapport d'activité

4.1. Cycle de développement

Pour le développement de notre site web, nous avons décidé de suivre un mode itératif.

Comme on peut le voir dans le schéma suivant, le développement de notre site web s'est divisé en plusieurs étapes :

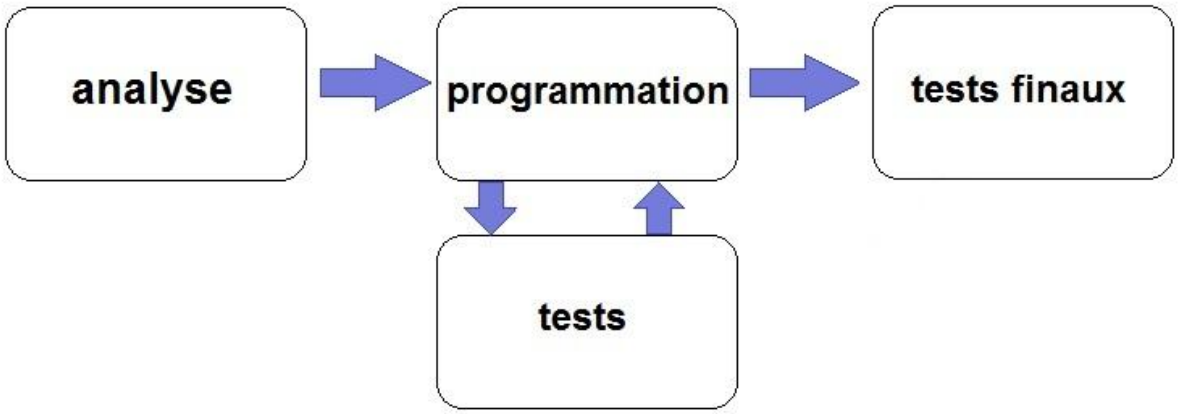


Figure 27 - Organisation du développement

-La première étape a été l'élaboration de tous les scénarios en utilisant des diagrammes adaptés. Nous avons de ce fait analysé les besoins des futurs utilisateurs du site web ainsi que les différents scénarios qui pouvaient avoir lieu. A partir de cette analyse, nous avons ainsi pu réaliser plusieurs diagrammes sur lesquels nous nous sommes appuyées tout au long de la réalisation du projet.

-La seconde étape fut la création des pages du site web. Nous avons donc bien suivi les diagrammes que nous avions produits afin de répondre aux besoins définis auparavant. Durant cette étape, nous avons effectué en parallèle des test des pages que nous étions en train de créer afin de pouvoir rectifier les erreurs le plus tôt possible et donc nous éviter d'avoir à reprendre tout le code de notre site lors de la phase de tests finaux.

-Après avoir créé tout le site web, la phase suivante concernait les tests finaux du site web. Cette phase est cruciale dans le développement car elle permet de vérifier l'absence d'erreurs qui pourraient gêner l'utilisateur. Par exemple, un mauvais affichage peut perdre l'utilisateur dans le site. Pendant cette période, nous avons donc entièrement testé le site web, nous avons testé chaque page du site ainsi que toutes les fonctionnalités que proposait notre site. Des personnes extérieures au projet ont aussi participé à la réalisation des tests afin que nous ayons le plus de cas de figure différents. En effet, ces personnes extérieures au projet et au monde de la programmation n'ont pas le même contexte perceptif que nous et donc ont

permis de constater des nouveaux éléments gênants que nous n'avions pas remarqués. Durant cette période, nous avons pu constater quelques erreurs, notamment des erreurs d'affichage que nous avons ensuite rectifié. Nous avons également facilité la navigation sur le site afin d'en améliorer la fluidité.

4.2. Planification

Au cours de la réalisation de notre site web, nous avons pu constater que nous n'avons pas réellement suivi le planning que nous avions réalisé visible dans l'annexe 6. En effet, au début de la réalisation de notre site web, nous nous sommes rendu compte que nous avions fait une erreur dans notre analyse de départ, ce qui voulait dire que la base de données que nous avions créée n'était pas correcte, nous avons donc dû recréer notre base de données. Notre erreur était d'avoir créé une table Arbre qui s'est révélée inutile et redondante puisque nous avions déjà l'identifiant de l'arbre dans la table Utilisateur.

De même, alors que théoriquement nous devions commencer la phase de test à partir du 5 décembre, nous n'avions, à ce moment-là, pas encore fini la phase de développement du site web. Nous n'étions donc pas dans les temps par rapport à notre calendrier prévisionnel.

Cette expérience nous a permis de prendre conscience que le planning prévisionnel que nous avions réalisé était loin d'être parfait. En effet, nous aurions dû accorder plus de temps au départ pour la conception des différents diagrammes et pour l'analyse des besoins des futurs utilisateurs.

De même, nous avons pu constater que la phase de développement du site web nous a pris plus de temps que prévu, ce qui veut dire que nous aurions dû prévoir plus de temps pour le développement du site web et peut être moins de temps sur la partie initialisation à PHP et MySQL voire supprimer carrément cette partie. En effet, nous n'avions pas pris en compte dans notre diagramme de Gantt la présence de la semaine web qui a rendu inutile la phase d'initialisation prévue.

Enfin, alors que nous avions prévu d'effectuer tous les tests après avoir entièrement programmé le site web, nous avons finalement décidé de tester le site web au fur et à mesure des pages créées. De cette façon, lors de la phase de tests, nous avons eu beaucoup moins d'erreurs à corriger que si nous n'avions pas effectué ses tests au fur et à mesure. De ce fait, la phase de tests finale nous a surtout permis l'optimisation ergonomique du site.

Le planning que nous aurions dû respecter et qui aurait été plus juste aurait donc été celui se trouvant en annexe 7.

4.3. Méthodes et outils de travail

Afin de réaliser un travail efficace, nous avons décidé de nous répartir les rôles et de diviser notre travail de façon équitable afin que nous fassions autant de choses l'une que l'autre. Ainsi, nous avons commencé tout d'abord par décider ensemble à quoi ressemblerait notre site web et nous avons défini les différentes fonctionnalités que nous voulions que notre site web propose. De ce fait, nous avons défini le cahier des charges et les diagrammes grâce à

Windesign, Modelio, et GanttProject. Puis nous avons divisé équitablement notre travail. Nous travaillions chacune de notre côté puis nous nous réunissions afin de mettre notre travail en commun. Lors de ces réunions, nous en profitions pour discuter des éventuels problèmes que nous avions par rapport à la programmation de certains aspects de notre site web et nous nous montrions mutuellement ce que nous avions fait. Pour échanger notre travail individuel à distance, notamment pendant les vacances ou les week-ends, nous avons utilisé notre messagerie électronique et Skype.

Bibliographie / sitographie

OUVRAGES CONSULTES

Mergnac, Marie-Odile, *Ma généalogie*, Archives & Culture, 2011, pages consultées (p.65-71)

SITES INTERNET CONSULTES

Adresse de la page	**Date de consultation**	**Type de site**	**Information recherchée**
www.siteduzero.com/	03/11/2012	Site communautaire	Gestion des formulaires
www.siteduzero.com/	09/11/2012	Site communautaire	Création du menu
https://kuler.adobe.com/	20/11/2012	Site de construction de chartes graphiques	Choix de couleurs pour le site
http://www.w3schools.com/html/html5_canvas.asp	08/12/2012	Site communautaire	Création des traits pour l'arbre généalogique

Conclusion

L'objectif principal de notre projet était de créer un site de généalogie participative. Pour cela, il était nécessaire de gérer la création, modification et suppression des arbres. De plus il était aussi nécessaire de permettre la recherche dans la base de données d'un arbre à partir du nom de sa racine. Nous avons géré les fonctionnalités de base d'un site classique en plus des fonctionnalités propres à la généalogie participative.

Notre site de généalogie participative répond bien au cahier des charges. Généal est finalement un site facile d'utilisation, fluide et agréable à utiliser. En effet, le minimalisme au niveau des couleurs rend le site attrayant, et adapté à tous types d'internautes. Les fonctionnalités demandées sont bien présentes sur le site.

Ce projet nous a permis d'améliorer nos compétences en programmation web, en algorithmie (grâce à la récursivité). Nous avons également appris à utiliser des fonctions php et JavaScript que nous n'avions jamais vu auparavant. De plus, au niveau de la planification, nous avons pu analyser les erreurs que nous avions faites au départ. De ce fait, nous saurons mieux gérer nos projets sur du long terme. Nous avons également appris à utiliser des logiciels que nous ne connaissions pas comme GanttProject.

Toutefois, afin d'optimiser le site, il y a plusieurs fonctionnalités que nous aurions pu ajouter. Par exemple, on pourrait proposer à l'utilisateur de construire son arbre sur un nombre de générations infini en lui proposant, lorsqu'il est sur son arbre, de pouvoir « centrer l'arbre sur un de ses arrières-grands parents ». L'arrière-grand parent concerné deviendrait alors la racine de l'arbre. On pourrait donc lui ajouter des parents, des grands-parents, etc. Une autre fonctionnalité qui aurait été possible d'ajouter est le fait qu'un utilisateur puisse faire plusieurs arbres généalogiques. Il aurait fallu ajouter une table Arbre car un simple attribut n'aurait plus suffit.

Annexes

Annexe 1 - Contrôleur mon arbre IV
Annexe 2 - Vue mon arbre V
Annexe 3 - Fonction supprimeIndividu() VI
Annexe 4 - Insérer mes parents dans l'arbre VI
Annexe 5- Arbre de résultat VII
Annexe 6 - Premier diagramme de gantt VIII
Annexe 7 - Diagramme de Gantt idéal IX

Annexe 1 - Contrôleur mon arbre

```php
<?php
include("./header.php");
include("../Modele/utilisateur.php");

if(isset($_SESSION['utilisateur'])){
        $Moi = Utilisateur::GetIndividuByAdmail($_SESSION['utilisateur']);

        //On traite les parents

        if($Moi->getnum_individu_mere() != null){
            $Mere = Individu::GetIndividuByNumIndividu($Moi->getnum_individu_mere());
        }

        if($Moi->getnum_individu_pere() != null) {
            $Pere = Individu::GetIndividuByNumIndividu($Moi->getnum_individu_pere());
        }

        //Grands-parents côté pere
        if(isset($Pere)) //si on a le pere on peut maintenant créer les grands parents du côté paternel
        {
            if($Pere->getnum_individu_mere() != null){
                $GdMere_P = Individu::GetIndividuByNumIndividu($Pere->getnum_individu_mere());
            }

            if($Pere->getnum_individu_pere() != null){
                $GdPere_P = Individu::GetIndividuByNumIndividu($Pere->getnum_individu_pere());
            }
        }


        //Grands-parents de la mere
        if(isset($Mere)) //si on a la mere on peut maintenant créer les grands parents du côté maternel
        {
            if($Mere->getnum_individu_mere() != null) {
                $GdMere_M = Individu::GetIndividuByNumIndividu($Mere->getnum_individu_mere());
            }

            if($Mere->getnum_individu_pere() != null) {
                $GdPere_M = Individu::GetIndividuByNumIndividu($Mere->getnum_individu_pere());
            }
        }

        //ARRIERES GRANDS PARENTS

        //Parents du grand pere paternel
        if(isset($GdPere_P)){
            if ($GdPere_P->getnum_individu_pere()!=null){
                $ArGdPere_GdPere_P = Individu::GetIndividuByNumIndividu($GdPere_P->getnum_individu_pere());
            }

            if ($GdPere_P->getnum_individu_mere()!=null){
                $ArGdMere_GdPere_P = Individu::GetIndividuByNumIndividu($GdPere_P->getnum_individu_mere());
            }
        }

        //Parents de la grand mere paternelle
        if (isset($GdMere_P)){
            if ($GdMere_P->getnum_individu_pere()!=null){
                $ArGdPere_GdMere_P = Individu::GetIndividuByNumIndividu($GdMere_P->getnum_individu_pere());
            }

            if ($GdMere_P->getnum_individu_mere()!=null){
                $ArGdMere_GdMere_P = Individu::GetIndividuByNumIndividu($GdMere_P->getnum_individu_mere());
            }
        }

        //Parents du grand pere maternel
        if (isset($GdPere_M)){
            if ($GdPere_M->getnum_individu_pere()!=null){
                $ArGdPere_GdPere_M = Individu::GetIndividuByNumIndividu($GdPere_M->getnum_individu_pere());
            }

            if ($GdPere_M->getnum_individu_mere()!=null){
                $ArGdMere_GdPere_M = Individu::GetIndividuByNumIndividu($GdPere_M->getnum_individu_mere());
            }
        }

        //Parents de la grand mère maternelle
        if (isset($GdMere_M)){
            if ($GdMere_M->getnum_individu_pere()!=null){
                $ArGdPere_GdMere_M = Individu::GetIndividuByNumIndividu($GdMere_M->getnum_individu_pere());
            }

            if ($GdMere_M->getnum_individu_mere()!=null){
                $ArGdMere_GdMere_M = Individu::GetIndividuByNumIndividu($GdMere_M->getnum_individu_mere());
            }
        }
        include("../Vue/mon_arbre_user.php");
    }
else {
        include("../Vue/mon_arbre_interdit.php");
    }

include("../footer.php");
?>
```

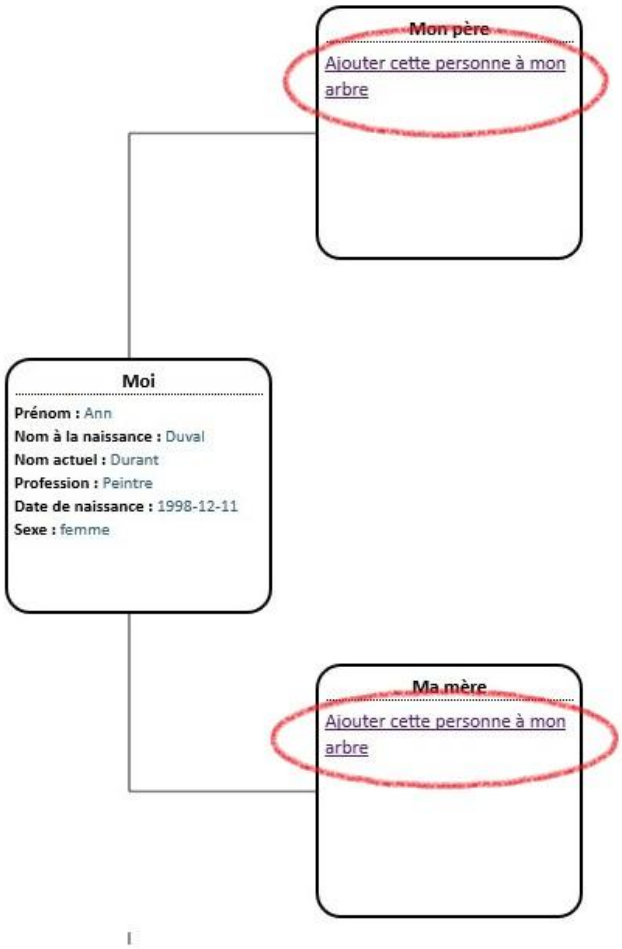
Annexe 2 - Vue mon arbre

```php
<!---------------------- PERE ---------------------->
    <div id="Pere" class="tree_box">
        <div class="tree_box_title">Mon père</div>
            <?php
                if(isset($Pere)) {
                    ?>
                    <div class="tree_box_informations">
                        <span class="tree_box_informations_title">Prénom : </span>
                        <span class="tree_box_informations_value"><?php echo $Pere->getprenom(); ?></span>
                    </div>
                    <div class="tree_box_informations">
                        <span class="tree_box_informations_title">Nom à la naissance : </span>
                        <span class="tree_box_informations_value"><?php echo $Pere->getnom_de_naissance(); ?></span>
                    </div>
                    <?php
                        if($Pere->getnom_actuel() != null) {
                            ?>
                        <div class="tree_box_informations">
                            <span class="tree_box_informations_title">Nom actuel : </span>
                            <span class="tree_box_informations_value"><?php echo $Pere->getnom_actuel(); ?></span>
                        </div>
                            <?php
                        }
                    ?>
                    <?php
                        if($Pere->getprofession() != null) {
                            ?>
                            <div class="tree_box_informations">
                                <span class="tree_box_informations_title">Profession : </span>
                                <span class="tree_box_informations_value"><?php echo $Pere->getprofession(); ?></span>
                            </div>
                            <?php
                        }
                    ?>
                    <?php
                        if($Pere->getdate_naissance() != "0000-00-00") {
                            ?>
                            <div class="tree_box_informations">
                                <span class="tree_box_informations_title">Date de naissance : </span>
                                <span class="tree_box_informations_value"><?php echo $Pere->getdate_naissance(); ?></span>
                            </div>
                            <?php
                        }
                    ?>
                    <div class="tree_box_informations">
                        <?php
                            if($Pere->getdate_mort() != "0000-00-00") {
                                ?>
                                <span class="tree_box_informations_title">Date de mort : </span>
                                <span class="tree_box_informations_value"><?php echo $Pere->getdate_mort(); ?></span>
                                <?php
                            }
                        ?>
                    </div>
                    <div class="tree_box_informations">
                        <span class="tree_box_informations_title">Sexe : </span>
                        <span class="tree_box_informations_value"><?php echo $Pere->getsexe(); ?></span>
                    </div>

                    <a href="../Controleur/modification_arbre_user.php?id=<?php echo $Pere->getnum_individu(); ?>">Modifier ses données</a>
                    <a href="../Controleur/suppression_personne_arbre_user.php?id=<?php echo $Pere->getnum_individu(); ?>">Supprimer cette personne</a>
                    <?php
                } else {
                ?>
                <a href="../Controleur/ajout_personne_arbre.php?idenfant=<?php echo $Moi->getnum_individu(); ?>&lien=pere">Ajouter cette personne à mon arbre</a>
                <?php
            }
        ?>
    </div>
```

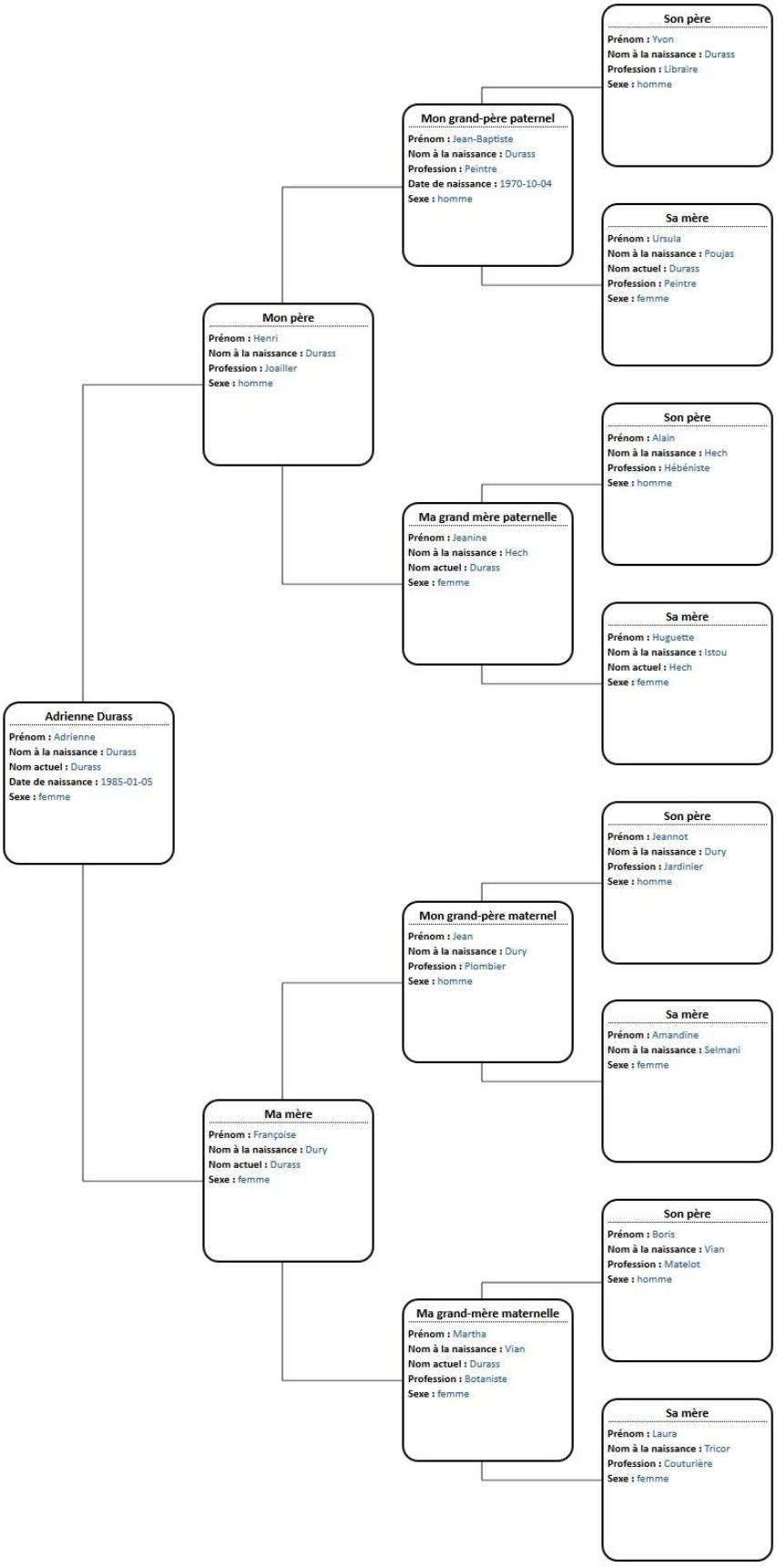
Annexe 3 - Fonction supprimeIndividu()

```php
//fonction qui supprime un individu donné et tous ses ascendants
public static function supprimeIndividu($num_individu)
{
    $Individu = Individu::GetIndividuByNumIndividu($num_individu);

    //si l'individu à supprimer possède un père ou une mère qui ont été saisis par l'utilisateur on passe dans cette boucle
    if($Individu->getnum_individu_pere() != null || $Individu->getnum_individu_mere() != null) {
        if($Individu->getnum_individu_pere() != null) { //si l'individu possède un pere
            Individu::supprimeIndividu($Individu->getnum_individu_pere()); //on applique la fonction supprimer au père pour supprimer les ancêtres de ce père
            }
        if($Individu->getnum_individu_mere() != null) { //si l'individu possède une mere
            Individu::supprimeIndividu($Individu->getnum_individu_mere()); //on applique la fonction supprimer au père pour supprimer les ancêtres de ce père
        }
    }

    //quand les parents de l'individu ont été supprimés
    $req = "DELETE FROM Individu WHERE num_individu = '". $Individu->getnum_individu() ."'";
    mysql_query($req) or die(mysql_error());

   //on remets à zéro les attributs num_individu_pere et num_individu_mere pour l'enfant de l'individu
    $req = mysql_query("SELECT * FROM Individu WHERE num_individu_pere = '$num_individu'");
    if($data = mysql_fetch_array($req)) {
        mysql_query("UPDATE Individu SET num_individu_pere = null WHERE num_individu = '{$data['num_individu']}'") or die(mysql_error());
    }

    $req = mysql_query("SELECT * FROM Individu WHERE num_individu_mere = '$num_individu'");

    if($data = mysql_fetch_array($req)) {
        mysql_query("UPDATE Individu SET num_individu_mere = null WHERE num_individu = '{$data['num_individu']}'") or die(mysql_error());
    }
}
```

Annexe 4 - Insérer mes parents dans l'arbre

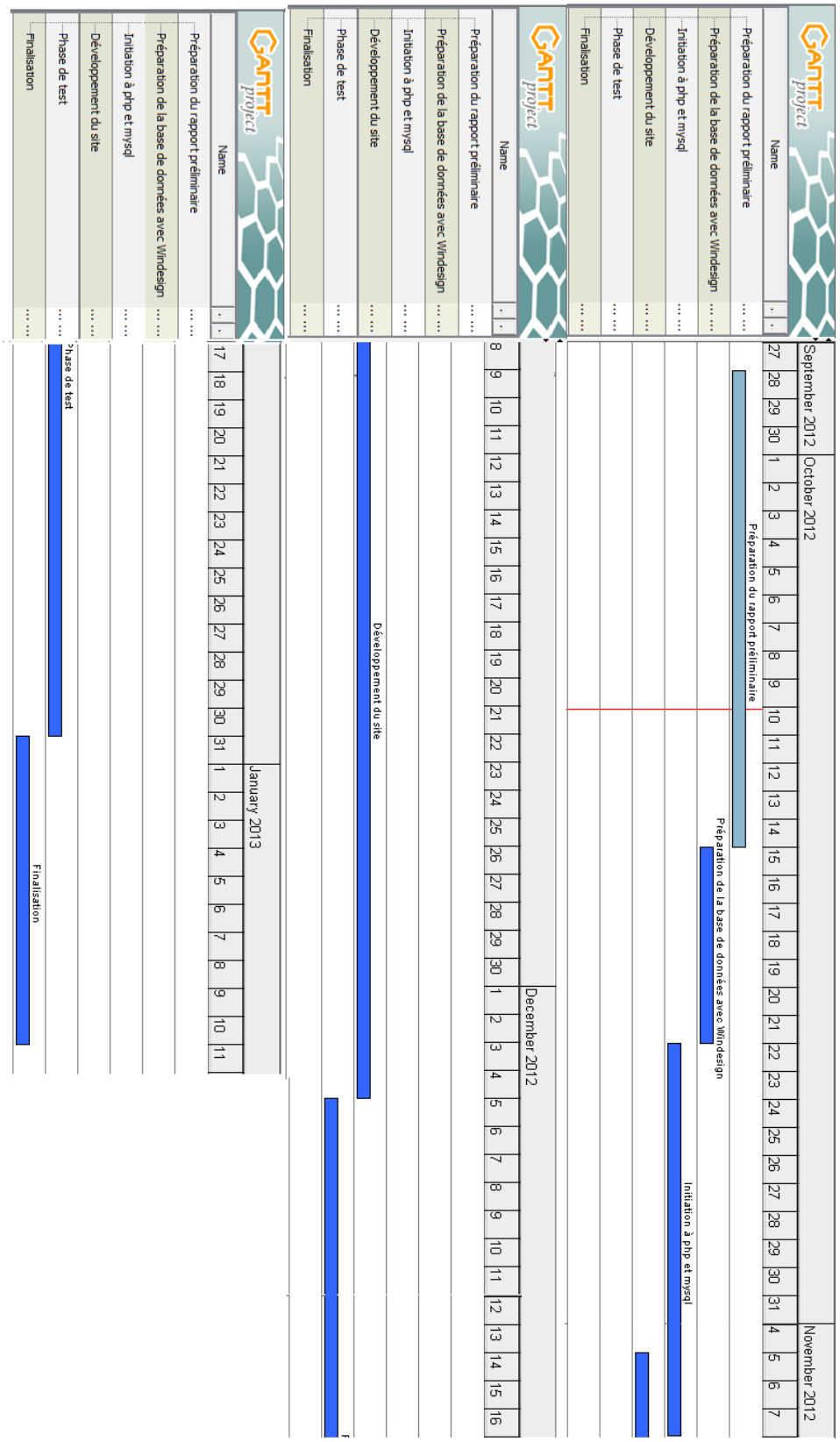


Annexe 5- Arbre de résultat



Annexe 6 - Premier diagramme de gantt

Annexe 7 - Diagramme de Gantt idéal



Résumé

Notre projet de réalisation de site de généalogie participative a pour but de permettre la création d'un arbre généalogique, son édition, son impression et son partage avec les autres membres du site. Les internautes doivent donc pouvoir s'inscrire sur le site, modifier leur compte, accéder à tout ce qui touche à leur arbre, et pouvoir rechercher dans les données du site des arbres déjà existants créés par d'autres utilisateurs.

Notre site a été développé en suivant le modèle MVC (modèle vue contrôleur). Il utilise en plus des langages basiques (HTML, CSS) le langage PHP et des requêtes SQL car notre site repose sur un outil PhpMyAdmin qui nous a permis de rassembler toutes les données relatives à notre site.

Mots clés

Généalogie participative, arbre généalogique, PHP, SQL, MVC, partage.

Summary

Our project is to realize a genealogy website and to allow its users to create their own genealogic tree. They should also have the possibility to edit it, print it and share it with the other users registered on the website. Net users can connect themselves to the website, modify their personal information, access to everything that concerns their tree, and research in the website's date trees that already exist and that were created by other users.

Our website was developed following the Model–View–Controller architecture. In addition to basic languages like html and css, our website was also developed using php and mysql queries since we decided to use phpmyadmin which helps us collect and save all the data needed for our website.

Key words

Participative genealogy, genealogy tree, PHP, SQL, MVC, sharing